



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,364	09/30/2003	Johannes Lauterbach	09700.0062	3223
22852	7590	09/18/2008		
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413			EXAMINER AUGUSTINE, NICHOLAS	
			ART UNIT 2179	PAPER NUMBER
			MAIL DATE 09/18/2008	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/676,364  
Filing Date: September 30, 2003  
Appellant(s): LAUTERBACH ET AL.

---

Peter C. Yi  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 06/24/2008 appealing from the Office action mailed 08/23/2007.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

\* Template Software SNAP Foundation Template Using the Snap Development Environment (SNAP), pages 2-2 - 2-5, 3-6 - 3-40, 4-2 - 4-14, 6-7 and 8-21, copyright 1998, released 1997

- \* Template Software SNAP Foundation Template Using the WFT Development Environment (ENV) chapter 3, copy right 1998, released 1997
- \* Template Workflow Template Process Foundation, Developing a WFT System (WFT) Whole Manual, copy right 1998, released 1997

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

***Claim Rejections - 35 USC § 102***

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-22 are rejected under 35 U.S.C. 102(b) as being anticipated by the Template Software product line.

The **Template** Software product line contains:

The SNAP programming language

The WorkFlow Template

The Web Component

These three-layered products work together.

The documentation sets for the products contains the following manuals.

**SNAP** released June 1997

SNAP Language Reference (Referred to as **REF** - Not used in this Office Action)

Using the SNAP Language (Referred to as **LANG** - Not used in this Office Action)

Using the SNAP Communication Component (Referred to as **COM**- Not used  
in this Office Action)

Using the SNAP Graphic User Interface Component (Referred to as **GUI**  
Not used in this Office Action)

Getting Started with SNAP (Referred to as **START** Not used in this Office Action)

Using the SNAP Display Editors (Referred to as **DISP** - Not used in this Office  
Action)

SNAP Class Library Reference (Referred to as **CLASS** - Not used in this Office  
Action)

Using the SNAP External Application Software Component (**EXT**- Not used in  
this Office Action)

Using the SNAP Development Environment (Referred to as **SNAP**)

SNAP Module Library Reference (Referred to as **MODU** -Not used in this Office  
Action)

Using the SNAP Permanent Storage Component (Referred to as **PERM**-  
Not used in this Office Action)

**Workflow** released September 1997

Developing a WFT Workflow System (Referred to as **WFT**)

Using the WFT Development Environment (Referred to as **ENV**)

WFT Library Reference (Referred to as **WFTLIB** - Not used in this Office Action)

### **Web Component**

Using the Web Component (Referred to as **WEB**- Not used in this Office Action)

### ***Training Guides***

SNAP Application Developer's Training Course (Referred to as **TRAINS** -

Modules 1 and 10 provided- Not used in this Office Action)

Workflow Template Training Course (Referred to as **TRAINW** - Section A)

Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press release announcing version 8.0 these considered prior art under ***in re Epstein*** 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date.

**As for independent claim 1**, Template teaches a computer program product, embodied in a tangible machine-readable information carrier, for developing applications, the computer program product being operable to cause data processing apparatus to interact with data conforming to a data model, the data model comprising (SNAP, page 4-2): a component class (SNAP, page 4-4); a model class associated with the component class (SNAP, page 4-4), *the model class including a model-class class and a model relation class, the model-class class including a model class attribute class, and the model relation class including a model relation role class* (SNAP, pages

4-10 – 4-14); a controller class associated with the component class, *the controller class including a context node class having a context attribute class, the context node class being associated with the model-class class and the model relation class, and the context attribute class being associated with the model class attribute class* (SNAP, pages 4-10 through 4-14); and a view class associated with the component class, *the view class including a user interface element class having a binding with either the context node class or the context attribute class* (SNAP, pages 4-10 – 4-14). (ENV, chapter 3)

As for dependent claim 2, Template teaches the computer program product of claim 1, wherein the data model further comprises a context element class that is a super class of the context node class and the context attribute class (SNAP, pages 3-21 and 3-22).

As for dependent claim 3, Template teaches the computer program product of claim 2; wherein the binding is associated with one of the context node class and the context attribute class using the context element class (SNAP, page 3-15).

As for dependent claim 4, Template teaches the computer program product of claim 1, wherein the association between the component class and the view class is an aggregation (SNAP, page 3-21 and 4-10 through 4-14).

As for dependent claim 5, Template teaches the computer program product of claim 1,

wherein the association between the component and the controller is an aggregation (SNAP, pages 3-21 and 4-10 through 4-14).

As for dependent claims 6 and 17, Template teaches the computer program product of claim 1 and corresponding system of claim 16, wherein the data model further includes an indicator that is used to determine a file border (SNAP, page 8-21).

As for dependent claims 7 and 18, Template teaches the computer program product of claim 1 and corresponding system of claim 16, wherein the data model further includes an indicator used to implement a platform-specific feature (SNAP, Chapter 8, page 8-26-UNIX and pages 2-4 through 2-5).

As for dependent claims 8 and 19, Template teaches the computer program product of claim 1 and corresponding system of claim 16, wherein the data model further includes an indicator representing translatable text (SNAP, page 6-10, Class Definition files (CD) more specific references to CD files throughout the reference).

As for dependent claims 9 and 20, Template teaches the computer program product of claim 1 and corresponding system of claim 16, wherein at least one of the associations in the data model is an aggregation, and wherein the data model further includes an indicator representing whether the aggregation is ordered (SNAP, Chapter 3, Object Model Editor, see pages 3-6 inheritance lines and page 3-9).

As for dependent claims 10 and 21, Template teaches the computer program product of claim 1 and corresponding system of claim 16, wherein the data model further includes an indicator representing a singular name (SNAP, page 3-11, New class – class name)

As for dependent claims 11 and 22, Template teaches the computer program product of claim 1 and corresponding system of claim 16, wherein the data model further includes an indicator representing whether an attribute is null able (SNAP, page 3-40, Attributes – Default).

As for dependent claim 12, Template teaches the computer program product of claim 1, wherein the data model further includes an unassociated class defining enumeration attributes representing allowed values of a specific enumeration type (SNAP, page 3-40, Attributes).

**As for independent claim 13,** Template teaches a computer program product, embodied in a tangible machine-readable information carrier, for developing applications, the computer program product being operable to cause data processing apparatus to (SNAP, page 4-2): *generate an instance of a model, the instance of the model including a model class instance and a model relation instance, the model class instance including a model class attribute instance, and the model relation instance including a model relation role instance; generate an instance of a controller, the*

*instance of the controller including a context node instance having a context attribute instance; generate an instance of a view, the instance of the view including a user interface element instance; associate the context node instance with the model class instance; associate the context node instance with the model relation instance; associate the context attribute instance with the model class attribute instance; and associate the user interface element instance with one of the context node instance and the context attribute instance (SNAP, pages 4-7 through 4-9).*

As for dependent claim 14, Template teaches the computer program product of claim 13, wherein the association between the controller instance and the context node instance is an aggregation (SNAP, page 3-21 and 4-10 through 4-14).

As for dependent claim 15, Template teaches the computer program product of claim 13, wherein the association between the model instance and the model class instance is an aggregation (SNAP, page 3-21 and 4-10 through 4-14).

**As for independent claim 16,** Template teaches a system for developing applications, the system comprising a repository including data conforming to a data model, the data model comprising (SNAP, page 4-2 and WFT, pages 2-2 through 2-3): *a component class; a model class associated with the component class, the model class including a model-class class and a model relation class, the model-class class including a model class attribute class and the model relation class including a model relation role class; a*

*controller class associated with the component class, the controller class including a context node class having a context attribute class, the context node class being associated with the model-class class and the model relation class and the context attribute class being associated with the model class attribute class; and a view class associated with the component class, the view class including a user interface element class having a binding with either the context node class and the context attribute class.* (Note the analysis of claims 1-3 above and WFT, chapter 2-4, 6-7 and pages 2-4 and 4-3).

---

#### **(10) Response to Argument**

Beginning on page 12 of Appellant's brief (hereinafter Brief); Appellant argues specific issues, which are accordingly addressed below.

A1. Appellants submit that the Examiner's use of multiple references in support of the rejection under 35 U.S.C. 102(b) is improper.

R1. The Examiner does not agree with Appellant. Stated in MPEP 2131.01 it is normally, only one reference should be used in making a rejection under 35 U.S.C. 102. However, a 35 U.S.C. 102 rejection over multiple references has been held to be proper when the extra references are cited to:

- (A) Prove the primary reference contains an "enabled disclosure;"
- (B) Explain the meaning of a term used in the primary reference; or

(C) Show that a characteristic not disclosed in the reference is inherent.

The references used in the rejection are manuals from The **Template** Software product line which contains:

The SNAP programming language

The WorkFlow Template

The Web Component

These three-layered products work together. Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press release announcing version 8.0 these considered prior art under *In re Epstein* 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date. Being that this is a single product offering conditions A and C are met from MPEP 2131.01 in that the primary reference (SNAP) is a piece of the entire software suite provided by Template Software and being that the other manuals (ENV and WFT) detailing different pieces of the entire software suite which SNAP is apart of (product line) would be inherent.

A2. Appellant argues that SNAP does not teach a "model class associated with the component class" and a "model class including a model-class class". In another words SNAP does not teach the claimed relationships (e.g. "associated with" and "including").

R2. Examiner does not agree. The exact names of classes are not used by SNAP but SNAP does not claim relationships. From pages 4-4 through 4-10 it is explained that there exist a "Display classes", "GUI class", "Domain classes", "Domain independent classes" and more "GUI classes" associated with a GUI class. A Display class functions as a "component class". A GUI Class functions as "model class". A Domain class along with more GUI classes and Domain independent class function as "model-class classes". The Display class is used as a starting point like a root node in a hierarchy, such that a Display class contains instructions to create and application in a window. Next a GUI class is used to decide which application will be placed in the window (e.g. bar chart). Then Domain class, Domain independent class, and further more GUI Classes (which make up small components to construct a desired application (e.g. scroll bar, title, etc) are used with the desired GUI Class (bar chart application) to create an application to be displayed in a window of the Display class. Therefore it is evident that SNAP shows a claim relationship which is identical to Appellants claimed language "model class associated with the component class" and a "model class including a model-class class".

#### **(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Nicholas Augustine/

Nicholas Augustine Examiner 2179

Conferees:

/Weilun Lo/

Supervisory Patent Examiner, Art Unit 2179

Ba Huynh

/Ba Huynh/

Primary Examiner, Art Unit 2179